IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 43, NO. 2, APRIL 1996

Single-Event-Effect Mitigation from a System Perspective

Kenneth A. LaBel and Michele M. Gates

Abstract— **With the sharp decline in the availability of radiation-hardened devices from manufacturers, as well as the desire to shrink power, weight, and volume of spacecraft systems, the use of devices that are susceptible to single-event effects (SEE's) has become commonplace. We present herein a perspective and user's tool for understanding SEE's and potential system-level mitigation techniques.**

I. Introduction

FOR many of today's spaceflight programs, spacecraft and spacecraft designers are being pushed to utilize enabling or emerging technology, including radiation-tolerant, or even radiation-soft devices, in order to meet performance constraints in small-volume and low-power, low-cost spacecraft. Examples of these technologies include GaAs integrated circuits (IC's), reduced voltage and submicron CMOS IC's, integrated optoelectronics at 850-, 1300-, and 1550-nm wavelengths, multichip modules (MCM's) in both two-dimensional and three-dimensional configurations, field-programmable gate arrays (FPGA's), solid-state power controllers (SSPC's), high-performance microprocessors, etc.

Why are these technologies enabling to spacecraft and designers? Their benefits may include higher gate densities, increased speed/performance, easier system development by using commercial off-the-shelf (COTS) development and test equipment, and in the case of commercial devices, decreased supplier lead times versus radiation-hardened (RH) devices. IC manufacturers are being driven by a commercial market, of which the space community is a very small portion. Because of this (and reduced Department of Defense RH efforts), these "soft" technologies must be evaluated to meet performance requirements of spacecraft, especially for the smaller, low-cost satellite programs. These technologies, however, may be vulnerable to the space SEE environment.

This paper shows that engineers and spacecraft programs need to be aware of the potential difficulties posed by the space-radiation environment and how to mitigate SEE's and their effects, not only on individual devices, but on system level designs as well. In the past, when most NASA engineers, project managers, and the like discussed the radiation hardness of their designs and spacecraft, they typically addressed total ionizing dose (TID) only. SEE has become important only recently. Still, many spacecraft designers and projects remain ignorant of SEE or do not understand the seriousness of the potential problem.

We divide this paper into the following sections: functional impacts of the SEE, historical perspective of the SEE, and example mitigation techniques. Please note that we generally discuss the impacts of single-event upsets (SEU's) to a system level, but discuss the single-event latch up (SEL) at a lower level.

II. Functional Impact of an SEE

An SEE may propagate through a circuit, subsystem, and system, thus making system-level impacts an important consideration [1]. The level of impact that the SEE has on its circuit, box, subsystem, etc., has an associated criticality, severity of occurrence, or risk, depending on the type and location of the SEE. For example, a device error or failure may have effects propagating to critical mission elements, such as a command error affecting thruster firing. There are also cases in which SEE's may have little or no observable effect at a system level. In fact, in most designs, there are specific areas that incur a less direct system impact from certain radiation effects. The data storage memory in a solid-state recorder (SSR), for example, may have error detection and correction coding (EDAC), which allows bit errors in the devices to appear transparent to the system performance. Fig. 1 presents a schematic of the different levels of design.

The levels of spacecraft design may be viewed as a layered series of increasing integration and complexity. The lowest level of spacecraft design is not directly shown in Fig. 1. This is the IC design, which is a subset of circuit design. For example, a portion of a circuit is placed inside a single IC. This design tier is called device level. The next level of design is the circuit level, which typically involves a quantity of IC's. The card-level design usually involves one or more circuits. Box-level design involves one or more cards, while subsystem design may involve multiple boxes. And finally, the spacecraft design requires multiple subsystems.

Because most SEE-inducing particles are not effectively attenuated by the use of shielding, design SEE tolerance requirements are not based on location in or on the vehicle. Instead, SEE tolerance requirements depend on the function or functions that the device performs. The result of a functional impairment on mission operations defines the system-level impact. Top-level functions, such as spacecraft safehold or science data processing, may involve several subsystems, such as the command and data handling (CADH) and attitude con-

Manuscript received June 30, 1995; revised December 8, 1995.
The authors are with NASA/GSFC, Code 735.1, Greenbelt, MD 20771 USA.
Publisher Item Identifier S 0018-9499(96)03339-4.

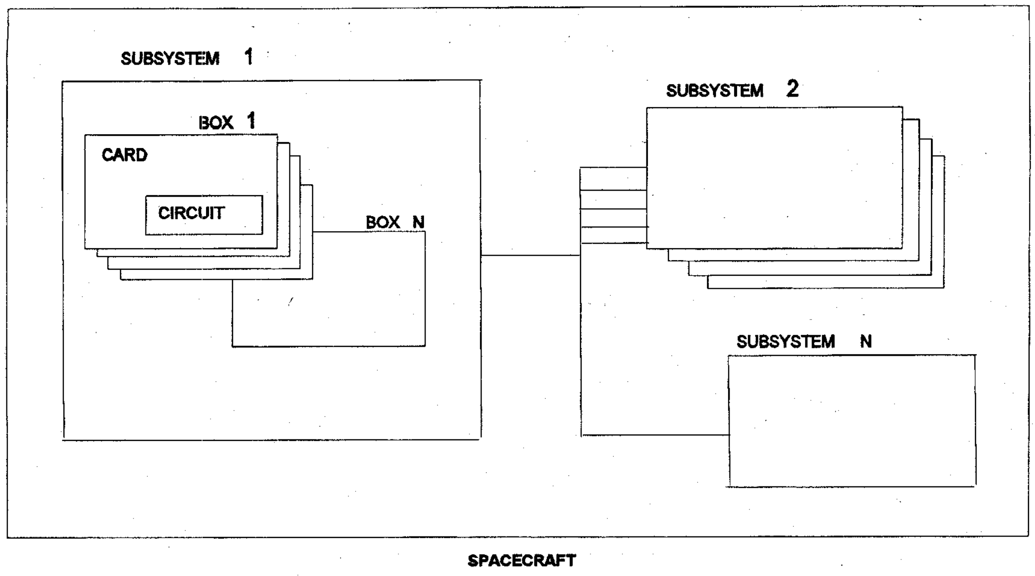

Fig. 1. Levels of a spacecraft design.

trol electronics (ACE) or instrument control subsystems. Many lower-level functions may be completely contained within one subsystem, such as an analog safehold function. SEE analysis is most effective when viewed from the perspective of the function(s) performed, rather than by physical boxes or subsystem architecture.

The criticality, or risk, of an operational impact to the system or spacecraft provides the foundation for setting a design requirement for SEE. System-level SEE requirements based on functional impact may be fulfilled through a variety of mitigation techniques, including the use of hardware, software, and device tolerance requirements. Unlike TID tolerances, SEE rates are not a mean-time-to-failure (MTTF) number, where the stopwatch begins at launch, but a probability that an SEE will occur within a known span of time. The cost, power, volume, performance, and availability of radiation-hardened devices often prohibit their use. Hardware or software design may serve as effective mitigation, but design complexity may present a problem. The most cost-efficient approach for meeting an SEE requirement may be an appropriate combination of SEE-hard devices and mitigation techniques.

For simplicity's sake, it is convenient to classify system-level effects into two general categories: those that affect data responses of a device and those that affect control of a device or system. Although there is some overlap between the two (an obvious example being a bit flip in a memory device that contains executable code for a processor), we may consider data errors to be those that occur in memory structures or data streams, and control errors to be in other hardware, such as microprocessors, power devices, or FPGA's.

All of the potential SEE mitigation methods may require that either additional hardware or software be added to the system design. The complexity and, in many cases, the increase in system overhead caused by the addition(s) are fairly linear with the power of the mitigation scheme.

III. System-Level SEE—A Historical Perspective

A. The First SEE on a Spacecraft—A System-Level Effect

Binder *et al.* [2] were pioneers in the field of spacecraft anomalies caused by an SEU event. They provided proof that certain anomalies observed in communication satellites could be traced to the interaction between a galactic cosmic ray and a sensitive area of a device such as a transistor. This interaction, in turn, caused bit flips to occur in a JK flip-flop device. Since their discovery, SEU anomalies have been documented yearly in IEEE Transactions on Nuclear Science and elsewhere (for example, [3]–[5]).

B. Ground Test and Simulation of System-Level or Propagated SEE's

Several groups have published information pertaining to either the simulation of SEU effects and their propagation to circuit and system level, as well as the performance of SEE ground testing on devices with the actual circuit design as used in a spacecraft system [6]–[17].

Newberry *et al.* [6]–[8] have been leaders in the area of SEU propagation. In particular, they have discussed the effects of radiation-induced input/output (I/O) transients or noise spikes on system performance, as well as that of very large scale integrated circuit (VLSIC) transients. The idea is that traditional bit flips in memory cells are not the only cause of SEU's on a system level, but SEU-induced voltage spikes occurring in logic or I/O devices also impact the system SEU rate and effects. This work was among the first to discuss transients and circuit-specific levels for defining SEU's (i.e., duration and amplitude constraints). For example, a 0.25-V spike of 5 ns in duration may or may not be observed by the following circuit elements.

Leavy *et al.* [9] have described the propagation of events inside a bulk CMOS microprocessor with SEU-hardened clocked flip-flops. In this instance, SEU-induced transients on the clock lines were shown to cause upsets to microprocessor operation. As a side note, Leavy, *et al.* solved this problem through a circuit redesign for their next foundry run of the microprocessor.

LaBel *et al.* [10], [11] have described the effects of transients in a fiber optic receiver photodiode as well as how this affects a system bit-error rate (BER) from both the physical link perspective and through higher layers of network protocol. This will be described below.

SEU-induced transients in analog devices have been reported by several organizations [12]–[14]. All of these references point out two facts. First, transients in devices such as a comparator or op amp may propagate to the digital electronics in the surrounding circuitry. Depending on the specific circuit designs, these transients may corrupt only a single telemetry sample or, in a worst-case scenario, cause system malfunction or failure. The second item was pointed out by Newberry [3] as well: The definition of an analog SEU phenomenon is specific to the interface circuitry surrounding the radiation-sensitive device.

Taking this one step further, Turflinger *et al.* [15], [16] have extensively delved into separating SEU's for conventional analog-to-digital converters (ADC's) into several categories. The two major categories are noise and offset errors that are analogous to Gaussian and non–Gaussian errors. Neither of these errors is fatal to the device itself, but both are capable of causing erroneous telemetry and misinterpretation by or impairment of the surrounding spacecraft systems.

McCarty *et al.* [17] also have explored an ADC. This ADC, however, was not a conventional successive-approximation register (SAR) or flash ADC, but a complex hybrid delta-sigma averaging ADC susceptible to both noise and offset errors, as well as control errors. These control errors are capable of affecting device operation and calibration. Furthermore, they hinder system performance in a space environment.

At this point, we have emphasized the effects of transient SEU's on system performance. This is not intended to slight digital SEU effects such as bit flips. These types of SEU's may propagate, for example, from a control or data register inside a microprocessor into operational performance of the circuit or system. A worst-case example may be the false commanding of critical hardware such as a thruster or pyrotechnic (explosive) device.

In some instances, one need not know what particular area of a device has seen an SEU, but how well the system mitigation design will work. NASA has been among the first to fly a commercial 32-bit microprocessor in a critical space application [18]. The Small Explorer Data System (SEDS) is a spacecraft command-and-data-handling subsystem for the Solar Anomalous Magnetospheric Particle Explorer (SAMPEX) mission at Goddard Space Flight Center (GSFC). Included as a critical portion of the SEDS is the recorder processor packetizer (RPP), an Intel 80386 microprocessor-based flight computer with 26.5 megabytes of solid-state data storage.

One of the design features of the SEDS is its built-in fault tolerance and its ability to recover from observed errors. This is accomplished via SEDS hardware watchdog circuitry (at multiple levels: circuit, board, box, etc.) as well as software health and safety tasks. To this end, an SEU test was performed on the RPP. SEU's were induced on the 80386 microprocessor family in order to verify the fault-tolerant capabilities of the SEDS [18]. The Brookhaven National Laboratories' tandem Tandem Van de Graaff accelerator was utilized for this purpose.

To summarize the SEDS ground SEE test results, several different errors were observed, including a halting of the RPP's operation and processor exceptions. All the SEE events were recoverable using planned mitigation techniques by the SEDS, as discussed in [18]. It should be noted that the SEDS has been performing flawlessly from the SEE mitigation perspective since its launch in July 1992.

IV. Sample System-Level Mitigation Techniques and Examples

A. *Classification of System-Level SEE's by Device Type*

Much as we partition SEE's into two areas, we may divide devices into two basic categories: those that are memory- or data-related devices, such as RAM's or IC's used in communication links or data streams, and those that are control-related devices, such as a microprocessor, logic IC, or power controller. That is not to say that there is no overlap between the two categories. For example, an error could occur in the cache region of a microprocessor and cause a data error, or a data SEU (bit flip) might occur in a memory device that contains an executable program potentially causing a control SEU.

B. *Mitigation of Memories and Data-Related Devices*

The simplest method of mitigating errors in memory/data stream is to utilize parity checks. This method counts the number of logic-one states, or "ones," occurring in a data path (i.e., an eight-bit byte or 16-bit word, etc.) [19]. Parity, usually a single bit added to the end of a data structure, states whether an odd or even number of ones was in that structure. This method detects an error if an odd number of bits is in error, but if an even number of errors occurs, the parity is still correct (i.e., the parity is the same whether zero or two errors occur). This is a detect-only method of mitigation and does not attempt to correct the error that occurs.

Another common error-detection-only method is called cyclic-redundancy check (CRC) coding [20]. This scheme is based on performing modulo-two arithmetic operations on a given data stream, then interpreting the result as a polynomial. The N data bits are treated as an $N - 1$ order polynomial. When encoding occurs, the data message is modulo-two divided by the generating polynomial. The remainder of this operation then becomes the CRC character that is appended to the data structure. For decoding, the new bit structure that includes the data and CRC bits is again divided by the generating polynomial. If the new remainder is zero, no detectable errors were observed. A commonly used CRC code, especially for mass storage such as tape recorders, is the CRC-16 code, which leaves a 16-bit remainder.

Hamming code is a simple block error encoding (i.e., an entire block of data is encoded with a check code) that will detect the position of a single error and the existence of more than one error in a data structure [19]. Hamming strategy essentially states that, if there are Q check bits generated using a parity-check matrix, then there is a syndrome represented by the Q-digit word that can describe the position of a single error. This is seen simply, for example, by having a syndrome (s), with $s = 000$ H being the no-error condition in a single byte, $s = 001$ being an error in bit one of the byte, and so on. By determining the position of the error, it is possible to correct this error. Most designers describe this method as single-bit correct, double-bit detect. This EDAC scheme is common among current solid-state recorders flying in space (for example, [3], [4]). When a system performs this EDAC procedure, it is called scrubbing (i.e., scrubbing of errors from clean or good data). An example would be an 80-bit-wide memory bus having a 72-bit data path and 8-bits of Hamming code. This coding method is recommended for systems with low probabilities of multiple errors in a single data structure (e.g., use only with a single-bit-error condition in a byte-wide data field).

Other block error codes, while beyond the scope of this paper in terms of operational description, provide more powerful error-correcting codes (ECC's). Among these, Reed–Solomon (R-S) coding is rapidly becoming widespread in its usage [21]. The R-S code is able to detect and correct multiple and consecutive errors in a data structure. An example [22] is known as (255 223). This translates to a 255 byte block having 223 bytes of data with 32 bytes of overhead at the end of the message. This particular R-S scheme is capable of correcting up to 16 consecutive bytes in error. This R-S encoding scheme is available in a single IC as designed by NASA VLSI Design Center [21]. A modified R-S scrubbing for an SSR has been performed in flight by software tasks as well [5].

Convolutional encoding [23], again outside the scope of operational description, is able to detect and correct multiple bit errors, but differs from block coding by interleaving the overhead or check bits into the actual data stream rather than being grouped into separate words at the end of the data structure. This style of encoding is typically considered for usage in communication systems and provides good immunity for mitigating isolated burst noise.

System-level protocol methods are best understood by illustration. The SEDS MIL-STD-1773 fiber-optic data bus has been successfully flying since July 1992 [8]. This system utilizes among its error-control features two methods of detection: parity checks and detection of a nonvalid Manchester encoding of data. This military standard has a system-level protocol option of retransmitting or retrying a bus transaction up to three times if the error detection controls are triggered. Thus, the error-detection schemes are via normal methods (parity or nonvalid signaling), while the error correction is via retransmission.

Retransmission of data on a communication link may be autonomously performed, as in the example above, or may be accomplished via ground intervention. For example, if data collected in an SSR shows an unacceptable BER during a pass or down-link transmission to a ground station, the station may then issue a command to the spacecraft requesting retransmission of all or a selected portion of that data.

TABLE I
SAMPLE EDAC METHODS FOR MEMORY OR DATA DEVICES AND SYSTEMS

EDAC Method	EDAC Capability
Parity	Single bit error detect
CRC Code	Detects if any errors occurred in a given data structure
Hamming Code	Single bit correct, double bit detect
RS Code	Correct consecutive and multiple bytes in error
Convolutional encoding	Corrects isolated burst noise in a communication stream.
Overlying protocol	Specific to each system implementation

All of the above methods provide ways of reducing the effective BER of data storage areas, such as SSR's, communication paths, or data interconnects. Table I summarizes sample EDAC methods for memory or data devices and systems.

C. Mitigation of Control-Related Devices

Although the above techniques are useful for data SEU's, they may also be applicable to some types of control SEU's as well (microprocessor program memory, again being an example). Other devices such as very large scale integration (VLSI) circuitry or microprocessors have more complex difficulties to be aware of. Potential hazard conditions include items such as the issuance of an incorrect spacecraft command to a subsystem or a functional interruption of the system operation. Microprocessors are among the many new devices that have hidden registers. These are registers that are not readily accessible external to the device (i.e., on I/O pins), but provide internal device control and whose SEU's could affect the device or system operation.

Microprocessor software typically has tasks or subroutines dubbed health and safety (H&S), which may provide some mitigation means directly applicable to SEE [24]. These H&S tasks may perform memory scrubbing utilizing parity or other methods on either external memory devices or registers internal to the microprocessor. The software-based mitigation methods might also use internal microprocessor timers to operate a watchdog timer (see below) or to pass H&S messages between spacecraft systems. A relevant example would be if the software provided a parity check on the stored program memory when accessing an external or internal device, such as an electrically erasable programmable read-only memory (EEPROM). If a parity error was detected on a program memory fetch, the software might then access (read) the memory location a second time, place the system into a spacecraft safing or safe-operations mode, or read the program from a redundant EEPROM.

Watchdog timers may be implemented in hardware or software or through a combination of both. Typically, watchdogs are thought of as an "I'm okay" method of error detection. That is, a message indicating the health of a device or system is sent from one location to another. If the message is not received by the second location within a set time period, a "time out" has occurred. In this instance, the system then may provide an action to the device, box, subsystem, etc. Watchdog timers may be implemented at many levels: subsystem-to-subsystem, box-to-box, board-to-board, device-to-device, etc. Watchdogs may be active or passive. These different types are best understood by example.

Example 1 is an active watchdog. Device A has to send an "I'm okay" pulse on a once-per-second basis to an independent device B. B, for example, is an interrupt controller for a microprocessor system. If A fails to send this pulse within the allocated time period, device B "times out" and initiates a recovery action, such as issuing a reset pulse, removing power, sending a telemetry message to the ground, placing the spacecraft into safing mode, etc. B's actions are very specific to each mission scenario and spacecraft mode of operation.

Example 2 is a passive watchdog timer. In spacecraft X's normal operating scenario, it receives up-link messages (commands, code patches, table loads, etc.) from the ground station every 12 h. There is a timer on-board the spacecraft that times out if no up-link is received within this 12-h (or perhaps, 24-h) time frame. The spacecraft then initiates an action, such as a switch to a redundant antenna or uplink interface, a power cycling of the uplink interface, etc. What makes this a passive watchdog is that no specific "I'm okay" needs to be sent between peers, but a monitoring of normal operating conditions is sufficient.

Redundancy between circuits, boxes, subsystems, etc. provides a potential means of recovery from an SEE on a system level. Autonomous or ground-controlled switching from a prime system to a redundant spare provides system designers an option that may or may not fit within mission-specific spacecraft power and weight restrictions. Redundancy between boxes is relatively straightforward; therefore, we present a lower system-level redundancy example. The MIL-STD-1773 fiber-optic data bus is a fully redundant bus with an A side and a B side. Redundancy, in this implementation, allows the system designer to automatically switch from the prime (A) side to the redundant (B) side for all transactions in case of a failed transmission on the A bus, or to retry on the B side in case of an A failure, or wait for a command to switch to B if the bus BER on the A side exceeds a specified limit, etc.

Operating two identical circuits with synchronized clocking is termed a lockstep system. One normally speaks of lockstep systems when discussing microprocessors [25]. Error detection occurs if the processor outputs do not agree, implying that a potential SEU has occurred. The system then has the option of reinitializing, safing, etc. It must be pointed out that, for longer spacecraft mission time frames, lockstep conditions for commercial devices must be well thought out. In particular, the TID degradation of the commercial devices must be examined for clock skew with increasing dose. This may potentially cause false triggers between two such devices if each responds to TID effects even slightly differently.

Voting is a method that takes lockstep systems one step further: having three identical circuits and choosing the output that at least two agree upon. Katz *et al.* [26] provide an excellent example of this methodology. They have proposed and SEU-tested a triple modular redundancy (TMR) voting scheme for FPGA's, i.e., three voting flip-flops per logical flip-flop. FPGA's, one should note, replace older LSI circuits in many systems by providing higher gate counts and device-logic densities. Thus, the IC count, as well as the physical space required for spacecraft electrical designs, may be reduced. The TMR scheme proposed does not come without an overhead penalty; one essentially loses over two-thirds of the available FPGA gate count by implementing this method.

The discussion of FPGA's brings out an interesting point: Systems are becoming increasingly more complex as well as integrated. Gate arrays, FPGA's, and application-specific IC's (ASIC's) are becoming increasingly more commonplace in electrical spacecraft designs. Liu and Whitaker [27] provide one such SEU-hardening scheme to provide SEU immunity in the custom IC design phase that is applicable to spacecraft designs. This method provides a logic configuration that separates the p-type and the n-type diffusion nodes within a memory circuit.

The use of good system-engineering practices for spacecraft contributes another means of SEU mitigation [28]. Items such as the utilization of redundant command structures (i.e., two commands being required to trigger an event, usually with each command having a different data value or address), increased signal power margins, and other fail-safe engineering techniques may aid an SEU-hardening scheme.

These and other system-level engineering practices usually allow designers to be innovative and discover sufficient methods for SEU mitigation as needed. The authors would like to point out that the greatest risk to a spacecraft system and conversely, the greatest challenge to an electrical designer, is having unknown device or system SEE characteristics.

D. Treatment of Destructive Conditions and Mitigation

In contrast to the previous discussion of SEU's, destructive SEE conditions may or may not be recoverable depending on the individual device's response. Hardening from the system level is difficult at best, and in most cases, not particularly effective.

This stems from several concerns. First, nonrecoverable destructive events, such as single-event gate rupture (SEGR) or burnout (SEB), require that redundant devices or systems be in place since the prime device fails when the event occurs. SEL may or may not have the same failure, with each malfunction response being very device specific. Microlatch [18], in particular, is difficult to detect since the device's current consumption may remain within specification for normal device operation. LaBel *et al.* [18] have demonstrated the use of a multiple watchdog time-out scheme as a potential mitigation. In this instance, the first level watchdog acts as an "I'm okay" within a local circuit board. If this watchdog is triggered, a reset pulse is issued to the local circuitry. If this trigger-reset scenario occurs N times continuously or fails to recover the board within X seconds, a secondary watchdog is triggered, which removes power from the board. Power is restored via a ground command. This SEDS system was successfully SEL tested at BNL.

For individual devices, a current limiting circuit that may also cycle power is often considered. The failure modes of this protection circuit, however, are sometimes worse than finding a less SEL-sensitive device (e.g., infinite loop of power cycling may occur). Hence, SEL should be treated by the designer on a case-by-case basis, considering the device's SEL response, circuit design, and protection methods. Please note that multiple latch-up paths are present in most circuits, each with a different current signature. This makes the designer's job difficult in specifying the required current limit.

A concern similar to microlatch exists if, for example, current limiting is performed on a card or higher integration level and not on an individual device. A single device might enter an SEL state with a current sufficient to destroy the device, but not at a high enough current level to trigger the overcurrent protection on a card or higher level. The key here is again to know the device's SEL current signatures for each of its latch-up paths.

One other, and more risky method of SEL protection due to its potential time lags to detect and recover, is best demonstrated by example. An ADC has a known SEL sensitivity. The device's current consumption is gathered periodically via a control processor. If the read current exceeds a specified limit, power cycling is performed. This method may also use either telemetry data points for ground intervention or a device's specific or internal calibration parameters to be successful [29]. Destructive SEE effects are discussed further in several papers in this issue.

E. Sample Methods of Improving Designs for SEE Performance

By changing the design of a circuit or certain circuit parameters, improved SEU performance may be gained. Marshall, *et al.* [30] and LaBel *et al.* [31] have demonstrated several ways of improving a fiber-optic link's SEU-induced BER. First is the selection of diode material (typically, III-V versus Si). The use of a III-V material results in a significantly smaller device-sensitive volume. A second way to reduce BER is by the selection of the method for received signal detection (edge triggered versus level sensitive) with a level-sensitive system being less SEU sensitive. A third scheme for BER reduction is to define a dynamic sensitive time window. This method essentially states that there are only certain time periods when the occurrence of a radiation-induced transient will have an observed effect. Last, by increasing the optical power margin, the BER is also reduced. These and similar techniques may apply to other designs as well.

F. Sample Method of Realistic SEE Risks and Usage

Deciding whether an SEE in a device has a risk factor that makes a device usable in spaceflight or not is complex at best. Many factors weigh into the concern: mission environment, device test data, modes of operation, etc. Several sample system issues may clarify the types of issues that are involved.

The SEDS RPP uses separate EEPROM's for its boot and application software storage on board the SAMPEX spacecraft [32]. These particular EEPROM's have shown a sensitivity to SEU's while being programmed, albeit not during read operations. In addition, stuck bits may occur during programming operations at linear-energy transfers (LET's) above Ni-58 (i.e., there is a low probability of occurrence in flight). Since its launch in July 1992, the application software EEPROM's have successfully been reprogrammed in flight twice, but with certain constraints. These mission-specific constraints include that fact that the time period for programming uses a relatively proton and heavy-ion flux-free portion of the orbit, and not allowing the boot EEPROM to be programmed during flight. Why was the risk taken? The SEDS verifies the newly programmed data by the use of a CRC code, as well as by ground-station activities prior to loading the new executable software for SEDS operations. If an incorrect byte was programmed into the device, this mitigation scheme would catch it. If a stuck bit is discovered in the EEPROM, a recovery option is built into the system that provides a memory mapping around the failed location. Last, since the actual time window during programming when the device is susceptible to error is very small, few, if any, particles capable of causing an anomaly are seen at the device. It should be noted, however, that the risk might be deemed unacceptable if continuous programming of the EEPROM was performed throughout the mission's orbit.

The SEDS system has previously been pointed out for its use of system-level error control in its fiber-optic data bus as well as for the use of Hamming code EDAC on its SSR [3], [14]. The SEDS system also has a multilayer system of watchdog timers that monitor system operation [32]. The layers are as follows.

1) A software task executing in the main spacecraft microprocessor that times out if a value is not passed by a second software task and that restarts the processor from a known state.
2) A programmable interrupt signal from the main spacecraft microprocessor that provides a reset pulse to an external timer circuit that times out if not written to within an N second window, causing a hardware reset pulse to occur to the processor.
3) If multiple reset pulses occur consistently, this same external timer circuit provides an H&S message to a secondary processor box whereupon the secondary takes action.
4) An "I'm okay" pulse between the prime and secondary processors that must occur once every X seconds upon which the secondary processor may remove/cycle power to the main processor or place the spacecraft in safehold until ground station intervention.
5) A multiday timer that places the spacecraft into safehold if proper system operations have not occurred within a 24-h period.

As one may observe, mitigation methods for the SEDS are performed on several levels: software, device, circuit/card, box, and subsystem/spacecraft. Also note the use of both active and passive watchdogs.

V. Summary

We have presented a sampling of information regarding SEE mitigation from the systems-design level. This has included defining functional impacts of SEE's, examples of spacecraft designs, potential methods of SEE mitigation, as well as an example of realistic risks in space utilization of a sensitive EEPROM. It is crucial that spacecraft programs and engineers become aware of the potential hazards created by SEE's as well as the options that exist to mitigate them.

Acknowledgment

The authors would like to acknowledge the insight provided by Dr. P. Marshall in numerous discussions prior to the drafting of this document.

References

[1] M. Gates, K. LaBel, J. Barth, P. Marshall, and A. Johnston, "Single event effect criticality analysis guidelines," currently being drafted for NASA.
[2] D. Binder, E. C. Smith, and A. B. Holman, "Satellite anomalies from galactic cosmic rays," *IEEE Trans. Nucl. Sci.*, vol. NS-22, pp. 2675–2680, Dec. 1975.
[3] K. A. LaBel, S. Way, E. G. Stassinopoulos, C. M. Crabtree, J. Hengemihle, and M. M. Gates, "Solid state tape recorders: spaceflight SEU data for SAMPEX and Toms/Meteor-3," in *Workshop Rec. 1993 IEEE Radiation Effects Data Workshop,* 1993, pp. 77–84.
[4] R. Harboe-Sorensen, E. J. Daly, and L. Adams, "Observation and prediction of SEU in Hitachi SRAM's in low altitude polar orbits," *IEEE Trans. Nucl. Sci.*, vol. 40, pp. 1498–1504, Dec. 1993.
[5] C. I. Underwood, R. Ecoffet, S. Duzellier, D. Falguere, " Observations of single-event upset and multiple-bit upset in non-hardened high-density SRAM's in the Topex/Poseidon orbit," *Workshop Rec. 1993 IEEE Radiation Effects Data Workshop,* pp. 85–92, 1993.
[6] D. M. Newberry, D. H. Kaye, and G. A. Soli, "Single event induced transients in I/O devices: A characterization," *IEEE Trans. Nucl. Sci.*, vol. 37, pp. 1974–1980, Dec. 1990.
[7] D. M. Newberry, "Single event upset error propagation between interconnected VLSI logic devices," in *RADECS 91: IEEE Proc.,* Sept. 1991, vol. 15, pp. 471–474.
[8] D. M. Newberry, "Investigation of single event effects at the system level," in *RADECS 93: IEEE Proc.,* Sept. 1993, pp. 113–120.
[9] J. F. Leavy, L. F. Hoffman, R. W. Shovan, and M. T. Johnson, "Upset due to a single particle caused propagated transient in a bulk CMOS microprocessor," *IEEE Trans. Nucl. Sci.*, vol. 38, pp. 1493–1499, Dec. 1991.
[10] K. A. LaBel, E. G. Stassinopoulos, and G. J. Brucker, "Transient SEU's in a fiber optic system for space applications," *IEEE Trans. Nucl. Sci.*, vol. 38, pp. 1546–1550, Dec. 1991.
[11] K. A. LaBel, P. W. Marshall, C. J. Dale, C. M. Crabtree, E. G. Stassinopoulos, and M. M. Gates, "SEDS MIL-STD-1773 fiber optic data bus: proton irradiation test results and spaceflight SEU data," *IEEE Trans. Nucl. Sci.*, vol. 40, pp. 1638–1644, Dec. 1993.
[12] R. Koga, S. D. Pinkerton, S. C. Moss, D. C. Mayer, S. LaLumondiere, S. J. Hansel, K. B. Crawford, and W. R. Crain, "Observation of single event upsets in analog microcircuits," *IEEE Trans. Nucl. Sci.*, vol. 40, pp. 1838–1844, Dec. 1993.
[13] R. Ecoffet, S. Duzellier, P. Tastet, C. Aicardi, and M. Labrunee, "Observation of heavy ion induced transients in linear circuits," in *Workshop Rec. 1994 IEEE Radiation Effects Data Workshop,* 1994, pp. 72–77.
[14] K. A. LaBel, A. K. Moran, D. K. Hawkins, J. A. Cooley, C. M. Seidleck, M. M. Gates, B. S. Smith, E. G. Stassinopoulos, P. W. Marshall, and C. J. Dale, "Single event effect proton and heavy ion test results for candidate spacecraft electronics," in *Workshop Rec. 1994 IEEE Radiation Effects Data Workshop,* 1994, pp. 64–71.
[15] T. L. Turflinger and M. V. Davey, "Transient radiation test techniques for high-speed analog-to-digital converters," *IEEE Trans. Nucl. Sci.*, vol. 36, pp. 2356–2361, Dec. 1989.
[16] T. L. Turflinger and M. V. Davey, "Single event effects in analog-to-digital converters: Device performance and system impact," *IEEE Trans. Nucl. Sci.*, vol. 41, pp. 2187–2194, Dec. 1994.
[17] K. P. McCarty, J. R. Coss, D. K. Nichols, G. M. Swift, and K. A. LaBel, "Single event effects testing of the crystal CS5327 16-bit ADC," in *Workshop Rec. 1994 IEEE Radiation Effects Data Workshop,* 1994, pp. 86–96.
[18] K. A. LaBel, E. G. Stassinopoulos, G. J. Brucker, and C. A. Stauffer, "SEU tests of a 80386 based flight-computer/data-handling system and discrete PROM and EEPROM devices, and SEL tests of discrete 80386, 80387, PROM, EEPROM and ASICS," in *Workshop Rec. 1992 IEEE Radiation Effects Data Workshop,* 1992, pp. 1–11.
[19] A. B. Carlson, *Communication Systems.* New York: McGraw-Hill, 1975.
[20] K. L Short, *Microprocessors and Programmed Logic,* 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1987
[21] W. K. Miller, private communication, 1995.
[22] R-S Encoder Data Sheet, NASA VLSI Design Center, 1994.
[23] W. L. Pritchard, H. G. Suyderhoud, and R. A. Nelson, *Satellite Communication Systems Engineering.* Englewood Cliffs, NJ: Prentice-Hall, 1993.
[24] R. J. Whitley, private communication, 1995.
[25] J. L. Kaschmitter, D. L. Shaeffer, N. J. Colella, C. L. McKnett, and P. G. Coakley, "Operation of commercial R3000 processors in the low earth orbit (LEO) space environment," *IEEE Trans. Nucl. Sci.*, vol. 38, pp. 1415–1428, Dec. 1991.
[26] R. Katz, R. Barto, P. McKerracher, R. Koga, "SEU hardening of field programmable gate arrays (FPGA's) for space applications and device characterization," *IEEE Trans. Nucl. Sci.*, vol. 41, pp. 2179–2186, Dec. 1994.
[27] M. N. Liu, S. Whitaker, "Low power SEU immune CMOS memory circuits," *IEEE Trans. Nucl. Sci.*, vol. 39, pp. 1679–1684, Dec. 1992.
[28] Engineering Directorate Electrical Design Guidelines, NASA/GSFC, 1991.
[29] S. K. Miller, private communication, 1994.
[30] P. W. Marshall, C. J. Dale, M. A. Carts, K. A. LaBel, "Particle-induced bit errors in high performance fiber optic data links for satellite data management," *IEEE Trans. Nucl. Sci.*, vol. 41, pp. 1958–1965, Dec. 1994.
[31] K. A. LaBel, D. K. Hawkins, J. A. Cooley, C. M. Seidleck, P. W. Marshall, C. J. Dale, M. M. Gates, H. S. Kim, and E. G. Stassinopoulos, "Single event effect ground test results for a fiber optic data interconnect and associated electronics," *IEEE Trans. Nucl. Sci.*, vol. 41, pp. 1999–2004, Dec. 1994.
[32] K. A. LaBel, unpublished, 1995.